DEPARTMENT OF STATISTICS
University of Wisconsin
1210 West Dayton St.
Madison, WI 53706

TECHNICAL REPORT NO. 1051r

April 16, 2002

Classification of Multiple Cancer Types by Multicategory Support Vector Machines Using Gene Expression Data

Yoonkyung Lee[1]
Department of Statistics
University of Wisconsin-Madison
`http://www.stat.wisc.edu/~yklee`

Cheol-Koo Lee
Molecular and Environmental
Toxicology Center
University of Wisconsin-Madison

[1] Research partly supported by NSF Grant DMS0072292 and NIH Grant EY09946. We have fixed some typos and added some references and comments on May 21. Section 2.2 has been revised on July 7.

Classification of Multiple Cancer Types by Multicategory Support Vector Machines Using Gene Expression Data

Yoonkyung Lee[†]
Department of Statistics
University of Wisconsin-Madison
yklee@stat.wisc.edu

Cheol-Koo Lee
Molecular and Environmental
Toxicology Center
University of Wisconsin-Madison

Abstract

Monitoring gene expression profiles is a novel approach in cancer diagnosis. Several studies showed that prediction of cancer types using gene expression data is promising and very informative. The Support Vector Machine (SVM) is one of the classification methods successfully applied to the cancer diagnosis problems using gene expression data. However, its optimal extension to more than two classes was not obvious, which might impose limitations in its application to multiple tumor types. In this paper, we analyze a couple of published multiple cancer types data sets by the multicategory SVM, which is a recently proposed extension of the binary SVM.

1 Introduction

Microarray gene expression technology has opened the possibility of investigating the activity of thousands of genes simultaneously. Gene expression profiles are the measurements of relative abundance of mRNA corresponding to the genes. Thus, gene expression profiles have potential as a medical diagnosis tool, since they sensitively reflect the state of a cell at the molecular level. In clinical practice, it is known that classification of cancer types primarily based on histological features has limitations due to their morphological similarity to other cancer types. Current diagnosis procedures typically involve a pathologist's interpretation of combination of analyses, without a single systematic test. Accurate diagnosis would be essential for the efficacy of therapies. Under the premise of gene expression patterns as fingerprints at the molecular level, systematic methods to classify tumor types using gene expression data have been studied recently, in an attempt to overcome the limitations of such conventional procedures. See Golub et al. (1999), Mukherjee et al. (1999), Dudoit et al. (2000), Furey et al. (2000), Khan et al. (2001), Yeo and Poggio (2001), and references therein.

Available training data sets (a set of pairs of a gene expression profile and the tumor type that it falls into) have a fairly small sample size, typically less than one hundred, compared to the number of genes involved. This poses an unprecedented challenge to some classification methodologies. The Support Vector Machine (SVM) is one of the methods successfully applied to the cancer diagnosis problem in the previous studies. In principle, it can handle input variables much larger than the

[†]Research partly supported by NSF Grant DMS0072292 and NIH Grant EY09946.

sample size since its solution is determined by the dual problem of size equal to the sample size. Although several extensions to the multiclass case have been proposed by Vapnik (1998), Weston and Watkins (1999), and Crammer and Singer (2000), its optimal extension was not obvious in relation to the theoretically best classification rule. Using SVMs, multiclass problems have been tackled by solving a series of binary problems instead, such as one-vs-rest schemes. In order to overcome possible limitations in its application to multiclass problems, the multicategory Support Vector Machine (MSVM), an optimal extension of the binary SVM, was proposed recently by Lee et al. (2001). In this paper, we apply the Multicategory Support Vector Machine (MSVM) to the leukemia data set in Golub et al. (1999) and the small round blue cell tumors (SRBCTs) of childhood data set in Khan et al. (2001). The classification results show that the MSVMs achieve perfect classification or near perfect classification in diagnosing blind test samples. Such classification accuracy is comparable to other methods. In addition to demonstrating the effectiveness of MSVMs for the diagnosis of multiple cancer types, we touch other issues related to the data analysis in this paper; the effect of data preprocessing, gene selection, and dimension reduction.

This paper is organized as follows. In Section 2, we briefly review the multicategory SVM and discuss how to assess the strength of prediction made by the MSVM and some heuristics to reject weak predictions, which may be important in clinical practice. The analysis of the two published data sets by MSVMs comprises Section 3, followed by concluding remarks and discussion at the end.

2 Multicategory Support Vector Machines

2.1 Brief review of Multicategory Support Vector Machines

The binary SVM paradigm has a nice geometrical interpretation of discriminating one class from the other by a separating hyperplane with maximum margin. See Boser et al. (1992), Vapnik (1995), and Burges (1998) for introductions to SVMs. Now, it is commonly known that the SVM paradigm can be cast as a regularization problem. See Wahba (1998) and Evgeniou et al. (1999) for details. In classification problems, we are given a training data set that consists of n samples, $(\mathbf{x}_i, y_i)$ for $i = 1, \cdots, n$. $\mathbf{x}_i \in R^d$ represents covariates or input vectors and y_i denotes the class label of the ith sample. In the binary SVM setting, the class labels y_i are either 1 or -1. Then SVM methodology seeks a function $f(\mathbf{x}) = h(\mathbf{x}) + b$ with $h \in H_K$, a reproducing kernel Hilbert space (RKHS) and b, a constant minimizing

$$\frac{1}{n}\sum_{i=1}^{n}(1 - y_i f(\mathbf{x}_i))_+ + \lambda \|h\|^2_{H_K} \tag{1}$$

where $(x)_+ = x$ if $x \geq 0$ and 0 otherwise. $\|h\|^2_{H_K}$ denotes the square norm of the function h defined in the RKHS with the reproducing kernel function $K(\cdot, \cdot)$, measuring the complexity or smoothness of h. For more information on RKHS, see Wahba (1990). λ is a tuning parameter which balances the data fit and the complexity of $f(\mathbf{x})$. The classification rule $\phi(\mathbf{x})$ induced by $f(\mathbf{x})$ is $\phi(\mathbf{x}) = sign[f(\mathbf{x})]$. The function $f(\mathbf{x})$ yields the level curve defined by $f(\mathbf{x}) = 0$ in R^d, which is the classification boundary of the rule $\phi(\mathbf{x})$.

For the multiclass problem, assume the class label $y_i \in \{1, \cdots, k\}$ without loss of generality. k is the number of classes. To carry over the symmetry in representation of class labels, we code each class label $\mathbf{y}_i$ as a k-dimensional vector with 1 in the jth coordinate and $-\frac{1}{k-1}$ elsewhere if it falls into class j for $j = 1, \cdots, k$. Accordingly, we define a k-tuple of separating functions $\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), \cdots, f_k(\mathbf{x}))$ with the sum-to-zero constraint, $\sum_{j=1}^{k} f_j(\mathbf{x}) = 0$ for any $\mathbf{x} \in R^d$. Note that the constraint holds implicitly for coded class labels $\mathbf{y}_i$. Analogous to the two-category case, we consider $\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), \cdots, f_k(\mathbf{x})) \in \prod_{j=1}^{k}(\{1\} + H_{K_j})$, the product space of k reproducing kernel Hilbert spaces H_{K_j} for $j = 1, \cdots, k$. In other words, each component $f_j(\mathbf{x})$ can be expressed as $h_j(\mathbf{x}) + b_j$ with $h_j \in H_{K_j}$. Unless there is compelling reason to believe that H_{K_j} should be different for $j = 1, \cdots, k$, we will assume they are the same RKHS denoted by H_K. Define Q as the k by k matrix with 0 on the diagonal, and 1 elsewhere. It represents the cost matrix when all the misclassification costs are equal. Let L be a function which maps a class label $\mathbf{y}_i$ to the jth row of the matrix Q if $\mathbf{y}_i$ indicates class j. So, if $\mathbf{y}_i$ represents class j, then $L(\mathbf{y}_i)$ is a k dimensional vector with 0 in the jth coordinate, and 1 elsewhere. The MSVM finds $\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), \cdots, f_k(\mathbf{x})) \in \prod_1^k(\{1\} + H_K)$, with the sum-to-zero constraint, minimizing

$$\frac{1}{n}\sum_{i=1}^{n} L(\mathbf{y}_i) \cdot (\mathbf{f}(\mathbf{x}_i) - \mathbf{y}_i)_+ + \frac{1}{2}\lambda \sum_{j=1}^{k} \|h_j\|_{H_K}^2 \tag{2}$$

where $(\mathbf{f}(\mathbf{x}_i) - \mathbf{y}_i)_+$ means $[(f_1(\mathbf{x}_i) - y_{i1})_+, \cdots, (f_k(\mathbf{x}_i) - y_{ik})_+]$ by taking the truncate function $(\cdot)_+$ componentwise, and $\cdot$ operation in the data fit functional indicates the Euclidean inner product. The classification rule $\phi(\mathbf{x})$ induced by $(f_1(\mathbf{x}), \cdots, f_k(\mathbf{x}))$ is $\phi(\mathbf{x}) = arg \max_j f_j(\mathbf{x})$. We can verify that the binary SVM formulation (1) is a special case of (2) when $k = 2$. Theoretically, the population version minimizer of the loss functional in (2) at $\mathbf{x}$ is proven to be the coded class label of the most probable class at $\mathbf{x}$ in Lee et al. (2001). This extends the binary case result that ordinary SVMs approximate the majority class label at $\mathbf{x}$, $sign(p_1(\mathbf{x}) - 1/2)$, asymptotically to the multiclass case. Here $p_1(\mathbf{x}) = P(Y = 1|X = \mathbf{x})$, where (X, Y) denotes a generic pair of a random sample from $P(\mathbf{x}, y)$. These implications are the main grounds to argue that solving a series of binary problems by the binary SVM may not be optimal for the original multiclass problems, and by contrast, the MSVMs implement the optimal classification rule asymptotically in a coupled fashion. So, for flexible reproducing kernel Hilbert space and appropriately chosen λ, the solution $\mathbf{f}(\mathbf{x})$ to (2) is expected to be close to the most probable class code.

The problem of finding constrained functions $(f_1(\mathbf{x}), \cdots, f_k(\mathbf{x}))$ minimizing (2) is then transformed into that of finding finite dimensional coefficients instead, with the aid of a variant of the representer theorem. It was shown in Lee et al. (2001) that to find $(f_1(\mathbf{x}), \cdots, f_k(\mathbf{x}))$ with the sum-to-zero constraint, minimizing (2) is equivalent to find $(f_1(\mathbf{x}), \cdots, f_k(\mathbf{x}))$ of the form

$$f_j(\mathbf{x}) = b_j + \sum_{i=1}^{n} c_{ij} K(\mathbf{x}_i, \mathbf{x}) \quad \text{for } j = 1, \cdots, k \tag{3}$$

with the sum-to-zero constraint only at $\mathbf{x}_i$ for $i = 1, \cdots, n$, minimizing (2). Omitting intermediate steps and introducing nonnegative Lagrange multipliers $\alpha_j \in R^n$, we get the following dual problem:

$$\min_{\alpha_j} L_D = \frac{1}{2}\sum_{j=1}^{k}(\alpha_j - \bar{\alpha})^t K(\alpha_j - \bar{\alpha}) + n\lambda \sum_{j=1}^{k} \alpha_j^t \mathbf{y}_{\cdot j} \tag{4}$$

$$\text{subject to} \qquad 0 \le \alpha_j \le L_j \qquad \text{for } j = 1, \cdots, k \tag{5}$$

$$(\alpha_j - \bar{\alpha})^t \mathbf{e} = 0 \qquad \text{for } j = 1, \cdots, k \tag{6}$$

where $L_j \in R^n$ for $j = 1, \cdots, k$ is the jth column of the n by k matrix with the ith row $L(\mathbf{y}_i)$, and similarly $\mathbf{y}_{\cdot j}$ denotes the jth column of the n by k matrix with the ith row $\mathbf{y}_i$. $\bar{\alpha}$ is the average of α_j's, and $\mathbf{e}$ denotes the vector of ones of length n. With some abuse of notation, the n by

n matrix $K \equiv (K(\mathbf{x}_i, \mathbf{x}_\ell))$. Once we solve the quadratic programming problem, the coefficients are found from the relation $\mathbf{c}_{\cdot j} = -\frac{1}{n\lambda}(\alpha_j - \bar{\alpha})$ for $j = 1, \cdots, k$. Here $\mathbf{c}_{\cdot j}$ is the jth column of the n by k matrix with the ijth entry being c_{ij}. b_j can be found from any of the examples with unbounded α_{ij} satisfying (5) strictly by the Karush-Kuhn-Tucker complementarity conditions. It is worth noting that if $(\alpha_{i1}, \cdots, \alpha_{ik}) = 0$ for the ith example, then $(c_{i1}, \cdots, c_{ik}) = 0$, so removing such example $(\mathbf{x}_i, \mathbf{y}_i)$ would not affect the solution at all. In the two-category SVM, those data points with nonzero coefficient are called support vectors. To carry over the notion of support vectors to the multicategory case, we define support vectors as examples with $\mathbf{c}_i = (c_{i1}, \cdots, c_{ik}) \neq 0$ for $i = 1, \cdots, n$. Thus, the multicategory SVM retains the sparsity of the solution in the same way as the binary SVM. For proofs and details of MSVMs, refer to Lee et al. (2001) and Lee et al. (2002).

As with other regularization methods, the efficiency of the method depends on the tuning parameters. So, choosing proper tuning parameters is important in the MSVM as well. An approximate leaving-out-one cross validation function, called Generalized Approximate Cross Validation (GACV) is derived analogously to the GACV proposed by Wahba et al. (2000) in the binary case. Its detailed derivation can be found in Lee et al. (2002). To define the GACV for the multiclass case, we need to introduce a prediction function $\mu(\mathbf{f})$ first, which is a critical element of the leaving-out-one lemma. μ truncates any component $f_j < -\frac{1}{k-1}$ to $-\frac{1}{k-1}$ and replaces the rest by $\dfrac{\sum_{j=1}^{k} I(f_j < -\frac{1}{k-1})}{k - \sum_{j=1}^{k} I(f_j < -\frac{1}{k-1})}\left(\dfrac{1}{k-1}\right)$ to satisfy the sum-to-zero constraint. If $\mathbf{f}$ has a maximum component greater than 1, and all the others less than $-\frac{1}{k-1}$, then $\mu(\mathbf{f})$ is given by a k-tuple with 1 on the maximum coordinate and $-\frac{1}{k-1}$ elsewhere. So, the function μ maps $\mathbf{f}$ to a class label representation when there is a class strongly predicted by $\mathbf{f}$. In contrast, if none of the coordinates of $\mathbf{f}$ is less than $-\frac{1}{k-1}$, μ maps $\mathbf{f}$ to $(0, \cdots, 0)$. Now, the GACV for multicategory SVMs is defined as

$$GACV(\lambda) = \frac{1}{n}\sum_{i=1}^{n} L(\mathbf{y}_i)\cdot(\mathbf{f}(\mathbf{x}_i) - \mathbf{y}_i)_+ + \frac{1}{n}\sum_{i=1}^{n}(k-1)K(\mathbf{x}_i, \mathbf{x}_i)\sum_{j=1}^{k} l_{ij}[f_j(\mathbf{x}_i) + \frac{1}{k-1}]_* c_{ij}(y_{ij} - \mu_{ij}(\mathbf{f})) \tag{7}$$

where $L(\mathbf{y}_i) \equiv (l_{i1}, \cdots, l_{ik})$, and $[x]_* = I(x \geq 0)$. In practice, one can choose the minimizer of GACV as appropriate tuning parameters without really doing the leaving-out-one crossvalidation (LOOCV), which might be computationally prohibitive for large samples. 5-fold or 10-fold CV based on misclassification counts is often used alternatively. However, in cancer diagnosis problems using gene expression patterns, we typically encounter data sets of small sample size. Thus LOOCV can be still a feasible tuning tool, and has been often adopted for the validation of classifiers in the previous studies.

2.2 Assessing Prediction Strength

This section concerns how to measure strength or confidence of a class prediction made by Support Vector Machines. Based on some reasonable confidence measures, we wish to reject any prediction weaker than a specified threshold. For classification methods that provide an estimate of the conditional probability of each class given $\mathbf{x}$, the issue of whether to reject a class prediction or not can be settled easily. Set a threshold for the prediction probability such that we make a prediction only when the estimated probability of the predicted class exceeds the threshold. We mentioned that SVMs target the representation of the most probable class itself without any probability estimate when flexible kernel functions are used. Linear SVMs do not provide probability estimates, either. The efficiency that SVMs enjoy in classification problems by targeting much simpler functions than the probability functions themselves, suddenly becomes a drawback in dealing with this issue. How to measure the strength of SVM prediction may seem to be an undue question since our goal in the original formulation of classification problems was simply to find a rule with the minimum error rate. However, in many applications such as medical diagnosis, making a wrong prediction could be more serious than reserving a call. For weakly diagnosed examples, getting further information from a specialized investigation or expert opinion would be an appropriate procedure for a more informative call.

There have been a couple of approaches to address this problem for SVMs in the binary case, and solving a series of binary SVMs in the multiclass case. The basic idea comes from the observation that SVM decision functions get small in magnitude near classification boundaries. So, it is natural to propose a confidence measure based on the evaluation of the SVM decision function, $f(\mathbf{x})$ at $\mathbf{x}$; the bigger $f(\mathbf{x})$ in the absolute value, the stronger the prediction. Mukherjee et al. (1999) suggested a confidence measure for an SVM output $f(\mathbf{x})$ and its induced class prediction in this direction. Assuming that $P(Y|X=\mathbf{x}) \approx P(Y|f(X)=f(\mathbf{x}))$, $P(Y=1)=P(Y=-1)$ and $P(f|Y=1)=P(-f|Y=-1)$, they asserted that a confidence of the SVM prediction f can be quantified using the relation that $P(Y|f) \propto P(f|Y)P(Y)$. For the estimation of $P(f|Y)$, they used leave-one-out estimates of f values from the training data set, along with the class label y of each example left out. However, for almost separable classification problems, the proposed computations can not be done properly due to the complete or quasi-complete separation. So, they heuristically defined the confidence level of an SVM prediction f as $1-\hat{F}(|f|)$ at the end of their applications, using the symmetry assumption that $P(f|Y=1)=P(-f|Y=-1)$. Here, $\hat{F}$ is the estimated cumulative distribution function of SVM outputs $|f|$. This heuristic measure implicitly assumes that the probability of a correct prediction given f depends only on the margin $sign(f)\cdot f=|f|$ and realizes the initial notion that the bigger the margin $|f|$, the stronger the prediction. Thus, the confidence level for an SVM output f is interpreted as the proportion of SVM predictions stronger than f. The proportion can be inferred from jackknife (LOO) estimates or any variants of crossvalidation of training samples. The confidence level seems to be a misnomer in that the smaller the confidence level, the stronger the prediction. In their application, the confidence level was limited to at most 95% to make a class prediction.

Here is a simple variant of the method for MSVMs. The MSVM output is a vector of the decision functions $(f_1,\cdots,f_k)$ evaluated at $\mathbf{x}$. A decision vector close to a coded class label can be considered as a strong prediction. The multiclass hinge loss in (2), $g(\mathbf{y},\mathbf{f}) \equiv L(\mathbf{y})\cdot(\mathbf{f}-\mathbf{y})_+$ sensibly measures the proximity between an MSVM decision vector and a coded class, reflecting how strong their association is in the classification context. It considers the sign and the magnitude of each coordinate of a decision vector simultaneously. Recall that given an MSVM decision vector $(f_1,\cdots,f_k)$, the predicted class is $arg\max_j f_j$. Analogous to the binary case, we assume that the probability of a correct prediction given $\mathbf{f}(\mathbf{x})=(f_1,\cdots,f_k)$ at $\mathbf{x}$, $P(Y=arg\max_j f_j|\mathbf{f})$ depends on $\mathbf{f}$ only through the multiclass hinge loss, $g(arg\max_j f_j,\mathbf{f})$ for the predicted class. Now, the smaller the hinge loss, the stronger the prediction. The strength of the MSVM prediction, $P(Y=arg\max_j f_j|\mathbf{f})$ can be inferred from the training data similarly by crossvalidation. For example, leave out the ith example $(\mathbf{x}_i,y_i)$, and get the MSVM decision vector $\mathbf{f}(\mathbf{x}_i)=(f_1,\cdots,f_k)$ at $\mathbf{x}_i$ based on the remaining samples. Then, get a pair of the loss, $g(arg\max_j f_j(\mathbf{x}_i),\mathbf{f}(\mathbf{x}_i))$ and the indicator of correct decision $I(y_i=arg\max_j f_j(\mathbf{x}_i))$ and repeat this calculation marching through the samples in the training data set. $P(Y=arg\max_j f_j|\mathbf{f})$, as a function of $g(arg\max_j f_j,\mathbf{f})$ can be estimated from the collection of pairs of the hinge loss and the indicator. If we further assume the complete symmetry of k classes, that is, $P(Y=1)=\cdots=P(Y=k)$ and $P(\mathbf{f}|Y=y)=P(\pi(\mathbf{f})|Y=\pi(y))$ for any permutation operator π of $\{1,\cdots,k\}$, it follows that $P(Y=arg\max_j f_j|\mathbf{f})=P(Y=$

$\pi(arg\max_j f_j)|\pi(\mathbf{f}))$. Consequently, under these symmetry and invariance assumption with respect to k classes, we can pool the pairs of the hinge loss and the indicator for all the classes, and estimate the invariant prediction strength function in terms of the loss, regardless of the predicted class. In almost separable classification problems, oftentimes we would see the loss values for correct classifications only, impeding the estimation of the prediction strength. Again, we can apply similar heuristics of predicting a class only when the corresponding loss is less than, say, the 95th percentile of the empirical loss distribution. This cautious measure will be exercised in the application following this section.

The second approach to reject a prediction by SVMs naturally arises in solving multiclass problems by binary classifiers in the one-vs-rest fashion. Breaking a multiclass problem into a series of unrelated binary problems is apt to yield unresolved calls such as non-membership prediction (it does not belong to any of the known classes) and conflicting prediction (it falls into more than one class). Though the possibility of having unresolved calls may not be desirable in general, such indecisive prediction does mean a weak call subject to rejection. Yeo and Poggio (2001) demonstrated the idea of rejecting the two kinds of predictions in a tumor classification problem with four types, using binary SVMs, and other methods in the one-vs-rest fashion. They could reject the predictions made on 5 test examples which indeed do not fall into any of the four classes. It is worth mentioning that the population version of this approach is equivalent to making a prediction only when the predicted class has more than a 50% chance of being correct. If a more stringent classification rule is necessary in making prediction for unseen examples, then one can use the nonstandard binary SVMs in Lin et al. (2002) which adjust the costs for two different types of misclassification to achieve a required accuracy.

3 Data Analysis and Results

3.1 Leukemia Data Set

The leukemia gene expression data set from Golub et al. (1999) is revisited. Classification of acute leukemias has been done based on subtle morphological differences under the microscope or results of histobiochemical and cytogenetic analyses. Recently, Golub et al. (1999) suggested gene expression monitoring for cancer classification, and they demonstrated this idea on the classification of two preclassified leukemias, ALL (acute lymphoblastic leukemia) and AML (acute myeloid leukemia). These two cancer types were identified based on their origins, lymphoid (lymph or lymphatic tissue related) and myeloid (bone marrow related), respectively. ALL could be further divided into B-cell and T-cell ALLs. Several classification methods were applied to the data set as a two-class (ALL/AML) problem. They include "weighted voting scheme" in the original paper, Golub et al. (1999) and the binary Support Vector Machine in Furey et al. (2000) and Mukherjee et al. (1999). Dudoit et al. (2000) compared the performance of several discrimination methods such as linear discriminant analysis, CART, and nearest neighbor classifiers, for three tumor classification problems. In the last paper, the leukemia set was analyzed as both two-class (ALL/AML) and three-class problems.

We consider the problem as a three cancer types classification problem. Table 1 shows the class distribution of the leukemia data set. 38 training examples were all from bone marrow samples while 24 test examples were from bone marrow, and the remaining 10 test examples from peripheral blood samples. It was mentioned in Golub et al. (1999) that the test set may be more heterogeneous than the training set due to the different origin of the samples and different sample preparation protocols. However, Dudoit et al. (2000) argued that the heterogeneity does not seem to be prominent. The number of variables (genes) in the study is 7129. Since those genes irrelevant to the class prediction would degrade the performance of classifiers, we need to select relevant genes first for the accuracy of prediction. Before variable selection, standardization of the variables is necessary. Standardization of each variable (gene) across samples is a usual way in any statistical analysis. However, standardization of each sample, in other words, each array, across genes is often adopted in gene expression analysis. Additional preprocessing steps were taken in Dudoit et al. (2000) for the data before the standardization: (i) thresholding (floor of 100 and ceiling of 16000), (ii) filtering (exclusion of genes with $\max / \min \leq 5$ and $\max - \min \leq 500$ across the samples), (iii) base 10 logarithmic transformation. The filtering resulted in 3571 genes. To see the effect of different preprocessing and standardization, we tried both (A) to standardize each gene, and (B) to preprocess the data first according as Dudoit et al. (2000), and to standardize each array, in the following analysis.

Selecting important variables (genes) out of 7129 would be a formidable task if we require learning classifiers with all the possible subsets of the variables. To circumvent the difficulty, simple prescreening measures were used to pick out relevant variables in the previous applications, Golub et al. (1999), and Dudoit et al. (2000). Since the multiclass problems are of current concern in this paper, we use the ratio of between classes sum of squares to within class sum of squares for each gene, following Dudoit et al. (2000). For gene ℓ, the ratio is defined as

$$\frac{BSS(\ell)}{WSS(\ell)} = \frac{\sum_{i=1}^{n}\sum_{j=1}^{k} I(y_i = j)(\bar{x}_{\cdot\ell}^{(j)} - \bar{x}_{\cdot\ell})^2}{\sum_{i=1}^{n}\sum_{j=1}^{k} I(y_i = j)(x_{i\ell} - \bar{x}_{\cdot\ell}^{(j)})^2} \tag{8}$$

where n is the training sample size, 38 in the leukemia data set, $\bar{x}_{\cdot\ell}^{(j)}$ indicates the average expression level of gene ℓ for class j samples, and $\bar{x}_{\cdot\ell}$ is the overall mean expression levels of gene ℓ in the training set. We pick genes with the largest ratios. For instance, Figure 1 depicts the expression levels of 40 important genes selected by (8) in a heat map for the training samples standardized according to (B). Each row corresponds to a sample, which is grouped into the three classes, and the columns representing genes, are clustered in a way the similarity within each class and the difference between classes are easily recognized. We can expect from the figure that the selected 40 genes would be informative in discriminating the three classes. Table 2 shows the list of the top 20 genes out of 40 genes which maximize class separation in terms of (8). These genes encode functional proteins responsible for transcription factor, development, metabolism and structure. Since B-cell ALL (ALLB) and T-cell ALL (ALLT) arise from the same origin, we expected that these classes show similar trend in gene expression. However, surprisingly, the inspection of 20 most informative genes revealed that gene expression patterns in ALLB are much closer to those in AML than ALLT. ALLT showed quite different gene expression behavior than others. The box plots in Figure 2 illustrate four different gene expression patterns chosen from the top ranked 20 genes in the training set. More than 10 genes showed patterns similar to (i) gene 1, which possibly

Table 1: Class distribution of the leukemia data set

Data set	ALL B-cell	ALL T-cell	AML	total
Training set	19	8	11	38
Test set	19	1	14	34
Total	38	9	25	72

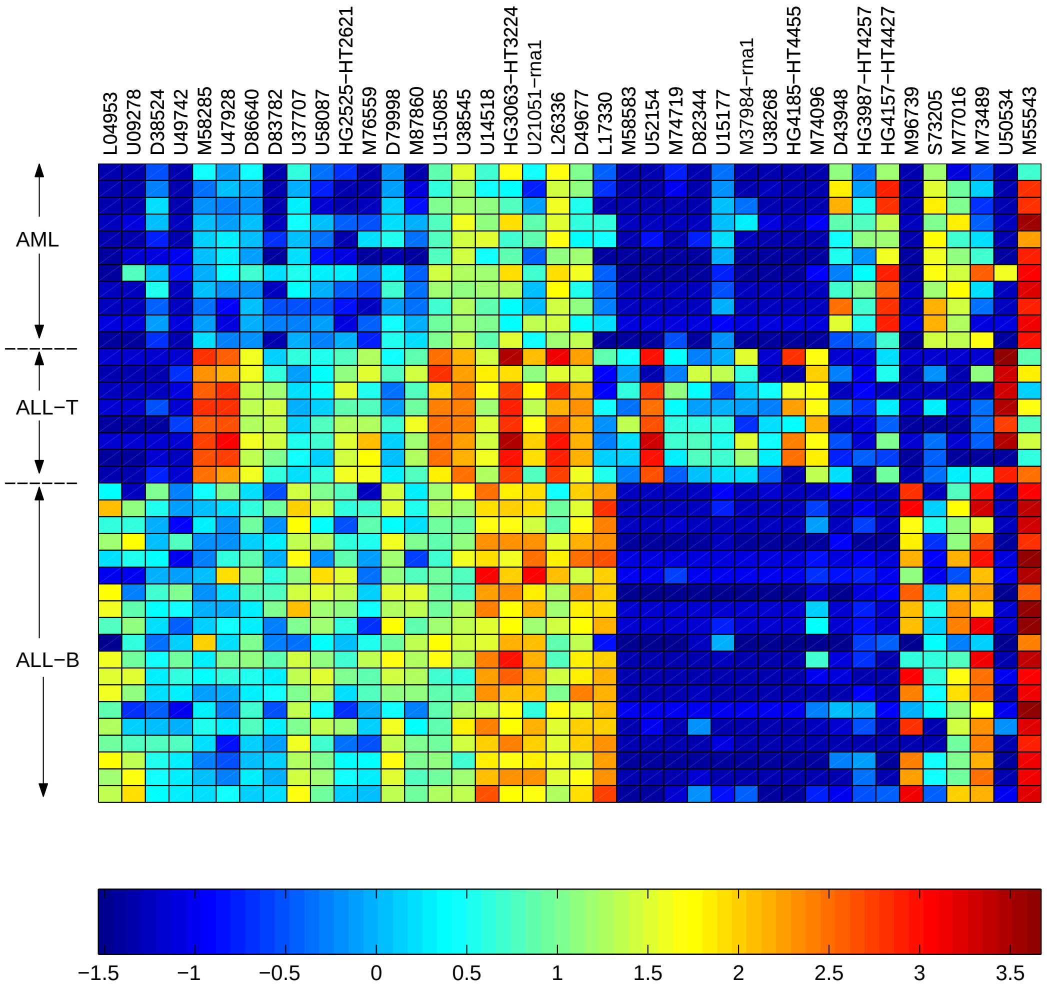

Figure 1: The heat map shows the expression levels of 40 most important genes for the training samples when they are standardized according to (B). Each row corresponds to a sample, which is grouped into the three classes, and the columns represent genes. The 40 genes are clustered in a way the similarity within each class and the dissimilarity between classes are easily recognized.

implies that ALLB might be closer to AML than ALLT. Whereas, only a few genes match with the patterns in (ii), (iii) and (iv). This is the reason as to why those genes listed as important predictors in distinguishing ALL from AML in Golub et al. (1999) do not overlap any of the genes in Table 2, other than the fact that different variable selection criteria and standardization procedures were used.

We apply the MSVM to the data with various numbers of genes, and different standardization steps. In addition, we compare the performance of the multicategory SVM classifiers for different choice of kernel functions, and tuning methods. For the choice of kernel function $K(\mathbf{x}_1, \mathbf{x}_2)$, the Gaussian kernel $\exp(-\frac{\|\mathbf{x}_1 - \mathbf{x}_2\|^2}{2\sigma^2})$ and the linear kernel $\mathbf{x}_1^t \mathbf{x}_2$ are considered. We compare two tuning methods; the leaving-out-one cross validation (LOOCV) based on misclassification counts and the Generalized Approximate Cross Validation (GACV). Table 3 summarizes the classification results. The first column is the number of variables included, with indication of the applied preprocessing

Table 2: 20 genes with the largest ratios in the leukemia data set

Rank	Orf	Gene	Description
1	M58285	*Hem-1*	Expressed only in cells of hematopoietic origin[1].
2	M58583	*Cerebellin 1 precursor (CBLN1)*	Also known as *Precerebellin 1*.
3	HG4157-HT4427	*Glycinamide Ribonucleotide Synthetase*	Metabolism.
4	M74096	*Acyl-CoA dehydrogenase, long-chain (Acadl)*	Beta-oxidation of fatty acids.
5	U52154	*Potassium inwardly-rectifying channel, subfamily J, member 5 (Kcnj5)*	Ion channel.
6	M74719	*Transcription factor 4 (TCF4)*	Also known as *Immunoglobulin transcription factor 2*. Plays a role in lymphoid development (B- and T-lymphocyte development)[2].
7	U15085	*Major histocompatibility complex, class II, DM beta (HLA-DMB)*	Antigen presentation. Immune-cell specific surface antigen.
8	U47928	*Protein "A"*	Unknown.
9	D82344	*Paired mesoderm homeobox 2b (PMX2B)*	Transcription factor. Also known as *Neuroblastoma paired-type homeo box gene (NBPHOX)*.
10	U50534	Unknown	Unknown.
11	U38545	*Phospholipase D1, phosphatidylcholine-specific (PLD1)*	Lipid metabolism. Phosphatidic acid, end product of this enzymatic activity, is a signaling molecule.
12	L04953	*Amyloid beta (A4) precursor protein-binding, family A, member 1 (X11) (Apba1)*	Putative function in synaptic vesicle exocytosis.
13	L17330	*Pre-T/NK cell associated protein (6H9A)*	Development.
14	D86640	*Src homology three (SH3) and cysteine rich domain (Stac)*	Signal transduction.
15	M96739	*Nescient helix loop helix 1 (Nhlh1)*	Transcription factor. Also known as *Hen1*. Plays an important role in growth and development.
16	U15177	*Alu*	Repetitive DNA sequence.
17	M37984	*Regulatory domain of cardiac troponin C*	Structural modulation.
18	D43948	*ch-TOG (for colonic and hepatic tumor over-expressed gene)*	Over-expressed in hepatomas and colonic tumors[3].
19	U14518	*Centromere protein A (17 kDa) (Cenpa)*	Component of centromere structure.
20	S73205	*Insulin activator factor (Insaf)*	Transcription factor binding at insulin control element (ICE).

[1] Hromas et al. (1991)

[2] Bain and Murre (1998)

[3] Charrasse et al. (1995)

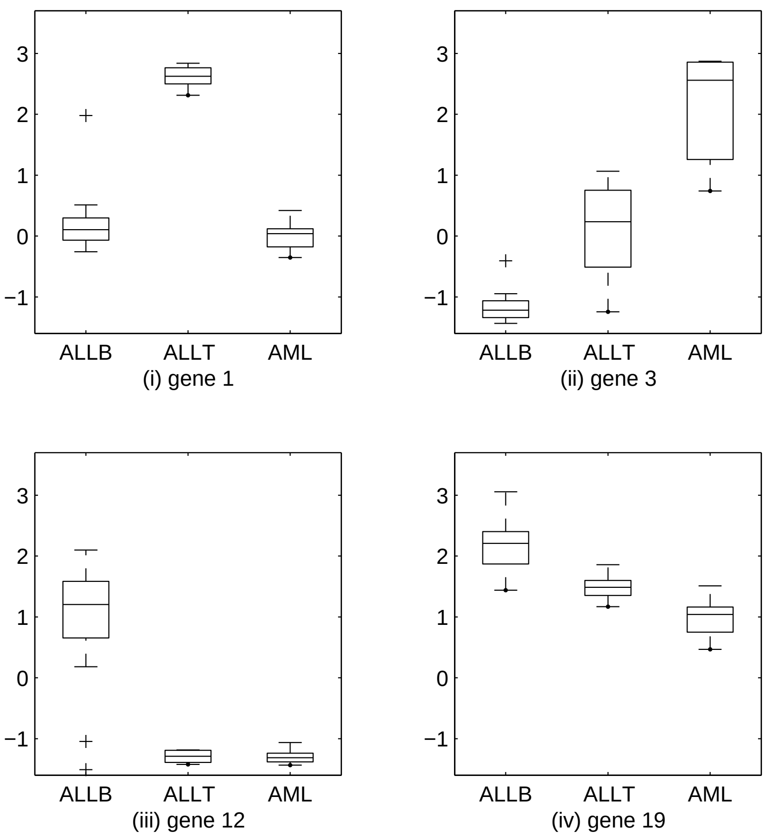

Figure 2: The box plots show four different gene expression patterns from the top ranked 20 genes in Table 2. A possible grouping of the genes depending on the expression patterns is (i) gene 1, 2, 4, 5, 6, 7, 8, 9, 10, 11, 14, 16, and 17, (ii) gene 3, and 18 (iii) gene 12, 13 and 15, and (iv) gene 19. Since gene 20 showed a slightly different pattern than the others, it was not included in the grouping. Only one representative gene from each group is shown in the four panels.

procedure (A) or (B). The values in the parentheses in the third column are optimal tuning parameters chosen by the specified method. A grid search was made for λ in the linear kernel case, and (λ, σ) jointly in the Gaussian kernel case. The optimal parameters are on log base 2 scale. Typically, the LOOCV in nearly separable classification problems does not have a unique minimum due to the non-convexity of the misclassification loss function. On the contrary, GACV gives a unique minimum, which is usually a part of the LOOCV multiple minima. Finally, the number of misclassified test samples, when the estimated classification rule is applied to 34 test samples, is in the last column, along with the misclassified test sample id's in the parentheses. The non integer values in the last column are due to the multiple minima. They are the averaged misclassification counts over all the tuning parameters chosen by LOOCV. The effect of the preprocessing steps can be read from the table. For the preprocessing (A), adding 50 more genes to the 50 important genes reduced the test error rate. Comparable or even smaller test error rate is achieved using only 40 genes when the data are preprocessed according to (B). Reducing the number of genes further down to 10, using (B) made the classification performance worse, which is not shown in the table. From the previous studies, the test error rates reported range from 0 to 5 out of 34. Note that they resulted from solving ALL vs AML binary problem, not to mention any difference in classification

Table 3: Classification results for the leukemia data set. The first column shows the number of genes fitted and the preprocessing method specified in parentheses. The second column indicates the kernel function employed in MSVMs and the third column identifies the tuning method with chosen tuning parameter(s) if unique, (multiple), otherwise. The counts of misclassified test samples are in the last column with the id's for the misclassified ones. When there are multiple equally good tuning parameters, the average performance was reported.

Number of genes	Kernel	Tuning $(\log_2 \lambda, \log_2 \sigma)$	Test error (test id)
50 (A)	Gaussian	GACV (-24,2.5)	4 (57, 60, 66, 71)
	Gaussian	LOOCV (multiple)	6 (53, 57, 60, 64, 66, 71)
	Linear	GACV (-13)	4 (57, 60, 66, 71)
	Linear	LOOCV (multiple)	4 (57, 60, 66, 71)
100 (A)	Gaussian	GACV (-22,2.6)	1 (66)
	Gaussian	LOOCV (-5, 2.8)	1 (66)
	Linear	GACV (-23)	2 (66, 71)
	Linear	LOO (multiple)	2.25 (57, 66, 67, 71)
40 (B)	Gaussian	GACV (-20,1.6)	1 (71)
	Gaussian	LOOCV (multiple)	0.8 (67,71)
	Linear	GACV (-13)	1 (71)
	Linear	LOOCV (multiple)	1 (71)

methods and preprocessing steps used in the papers. So, the test error rates from other studies may not be directly comparable to that of the multicategory SVM presented here. However, considering that multiclass problems are harder than binary problems, the performance of the MSVM in this three-class problem seems encouraging. Also, test samples frequently misclassified in this analysis, primarily intersect with those reported in the previous analyses. The Gaussian kernel function seems to give a slightly better result than the linear kernel function, but there is no significant difference. In terms of the test error rates, none of the tuning methods, GACV and LOOCV, gives a dominantly better result than the other.

Figure 3 shows the evaluations of the three components f_1, f_2, and f_3 for 34 test samples, when the Gaussian kernel function was used with the GACV tuning method for the data preprocessed as (B). Recall that the population version of (f_1, f_2, f_3) in the MSVM is the representation of the most probable class, and accordingly, we predict class j with the maximum component f_j at $\mathbf{x}$. In this example, the class codes, $(1, -1/2, -1/2)$, $(-1/2, 1, -1/2)$ and $(-1/2, -1/2, 1)$ correspond to ALL B-cell lineage, ALL T-cell lineage, and AML, respectively. Each test sample is drawn in a different color depending on the class that it falls into; ALL B-cell in blue, ALL T-cell in red, and AML in yellow. We can see from the figure that overall the estimated (f_1, f_2, f_3) approximates the corresponding class code and their proximity to the ideal class code varies along the test samples. Out of 34, there is one misclassified sample located at 56, which is from ALL B-cell, but classified into ALL T-cell. The test samples were rearranged in the figure, in order that samples from the same class cluster together. So, each value in the horizontal axis does not match with the test id in the original data set. The test id of the misclassified sample is 71.

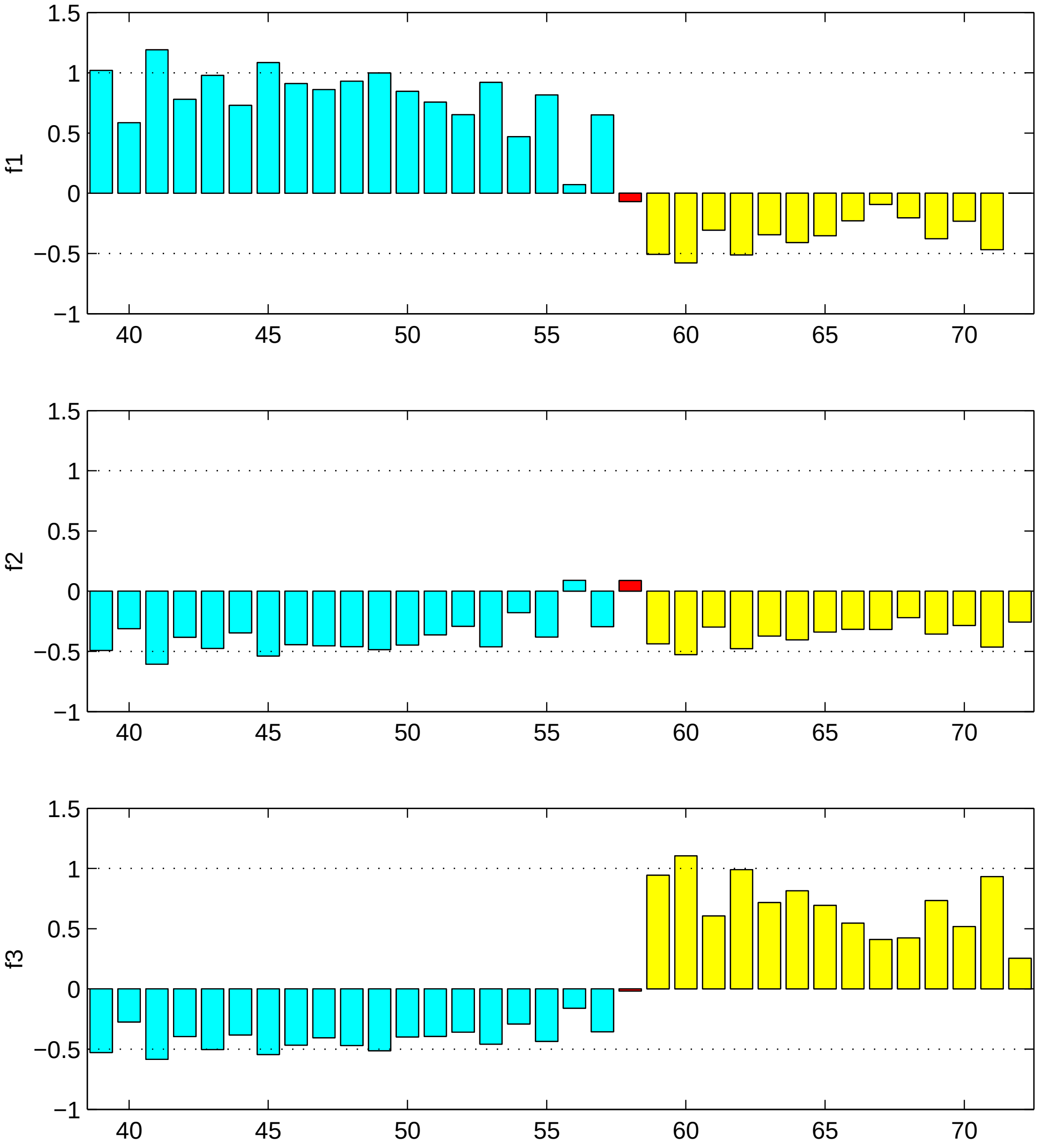

Figure 3: Evaluations of the three components (f_1, f_2, f_3) for 34 test samples. The class codes, $(1, -1/2, -1/2)$, $(-1/2, 1, -1/2)$ and $(-1/2, -1/2, 1)$ correspond to ALL B-cell lineage, ALL T-cell lineage, and AML, respectively. The true test example class is indicated by colors; ALL B-cell in blue, ALL T-cell in red, and AML in yellow. There is one misclassified sample located at 56.

3.2 Small Round Blue Cell Tumors of Childhood

Khan et al. (2001) classified the small round blue cell tumors (SRBCTs) of childhood into 4 classes; neuroblastoma (NB), rhabdomyosarcoma (RMS), non-Hodgkin lymphoma (NHL) and the Ewing family of tumors (EWS) using cDNA gene expression profiles. The data set is available from `http://www.nhgri.nih.gov/DIR/Microarray/Supplement/`. 2308 gene profiles out of 6567 genes are given in the data set after filtering for a minimal level of expression. The training set consists of 63 samples falling into 4 categories each, while the test set contains 20 SRBCT samples and 5 non SRBCTs (2 normal muscle tissues and 3 cell lines including an undifferentiated sarcoma, osteosarcoma, and a prostate carcinoma). Table 4 shows the distribution of the four distinct tumor categories in the training set and the test set. Note that Burkitt lymphoma (BL) is a subset of NHL. Khan et al. (2001) successfully diagnosed the tumor types into four categories using Artificial Neural Networks. Also, Yeo and Poggio (2001) applied k Nearest Neighbor (kNN), weighted voting and linear SVM in one-vs-rest fashion to this four-class problem, and compared the performances of these methods when they are combined with several feature selection methods for each binary classification problem. It was reported that mostly SVM classifiers achieve the smallest test error and LOOCV error when 5 to 100 genes (features) are used. For the best results shown in the paper, perfect classification was possible in testing the blind 20 samples as well as in crossvalidating 63 training samples with one training example left out each time. Yeo and Poggio (2001) did not tell how the classifiers with a tuning parameter such as kNN and SVM were tuned. However, it is clear from the context that the LOOCV errors reported in the paper refer to the misclassification counts over 63 training samples, each of which is left out to validate a classifier tuned and trained from the remaining 62 examples. Not to be confused, we refer to LOOCV tuning error whenever LOOCV is used as a tuning measure in this section. Since the one-vs-rest scheme needs four binary classifiers in this problem, the maximum number of distinct features used in learning a complete classification rule is four times the number of features for each binary classifier.

For comparison, we apply the MSVM to the problem using 20, 60, 100 genes. We take logarithm base 10 of the expression levels and standardize arrays before applying the classification method. Table 5 shows the list of top 20 genes. Most of genes were consistently selected from the top 96 genes used for the analysis in Khan et al. (2001). However, the list includes 4 additional genes, which are *neurofibromin 2*, *Isg20*, *cold shock domain protein A*, and *WASP*, and their biological functions are poorly characterized. Table 6 is a summary of the classification results by MSVMs. The Gaussian kernel function was our choice of kernel function in the analysis. Though the previous studies showed that linear classifiers are good enough to achieve almost perfect classification, we find that flexible basis functions such as the Gaussian kernel are particularly effective for multiclass problems. The classification results with the linear kernel function are not shown in the table, but it was observed that linear MSVMs achieve similar performances as Gaussian MSVMs although their evaluated decision vectors are less specific to the class representation than those of the Gaussian kernel. The second column indicates the optimal tuning parameters pair λ and σ on log 2 scale

Table 4: Class distribution of SRBCTs data set

Data set	NB	RMS	BL	EWS	total
Training set	12	20	8	23	63
Test set	6	5	3	6	20
Total	18	25	11	29	83

chosen by the GACV tuning measure (7). In fact, the LOOCV tuning error as a function of the tuning parameters was zero at multiple minima. The phenomenon that LOOCV tuning error has multiple minima while the multiple minima include the optimal tuning parameters given by GACV was observed in this experiment as well. The zero LOOCV tuning errors imply that the classification task is not challenging at all. The number of Support Vectors (SVs) in the third columns indicates how many samples out of 63 have nonzero coefficients in the expression of the solution (3). Removing non Support Vectors does not change the solution, and the number of SVs is related to the fraction of the training data near the classification boundary induced by the SVM. We observe from the table that a large number of features involved tend to produce a large number of SVs. It seems due to the sparsity of the data in a high dimensional space. The proposed MSVMs are crossvalidated for the training set in leaving-out-one fashion, with zero error attained for 20, 60, and 100 genes, as shown in the fourth column. The last column shows the final test results. Genes are selected according to the ratio of between class variability relative to within class variability in (8). Using the top ranked 20, 60, and 100 genes, the MSVMs correctly classify 20 test examples. With all the genes included, one error occurs in LOOCV and the misclassified example is identified as EWS-T13, which was reported to occur frequently as an LOOCV error in Khan et al. (2001) and Yeo and Poggio (2001). The test error using all genes varies from 0 to 3 depending on tuning measures. The MSVM tuned by GACV gives 3 test errors while LOOCV tuning gives 0 to 3 test errors.

Perfect classification in crossvalidation and testing with high dimensional inputs, suggests a possibility of a compact representation of the classifier in a low dimension. The main obstacle of analyzing high dimensional data like gene expression data is that we are not capable of visualizing the raw data in their original space, and consequently it is hard to make judicious calls in fitting and assessing models. However, such high dimensional data oftentimes reside in a low dimensional subspace. Using dimension reduction techniques such as the principal component analysis, we can visualize the data approximately in a much lower dimension than that of the original space. Figure 4 displays the three principal components of the top 100 genes in the training set as circles. Squares represent the corresponding three principal coordinates of the test set when we apply the linear combinations obtained from the training set to the test samples. Different colors identify four different tumor types; EWS in blue, BL in purple, NB in red, RMS in green, and non SRBCT in cyan. Notice that the principal coordinates of 5 non SRBCTs in the test set land on ‘no man’s land’, encircled by the samples from the four known classes. It clearly shows that three linear combinations of the 100 gene expression profiles are informative enough to differentiate 4 tumor types. The three principal components contain total 66.5% variation of 100 genes in the training set. They contribute 27.52%, 23.12% and 15.89%, respectively and the fourth component not included in the analysis explains only 3.48% of variation of the training data. With the three principal components (PCs) only, we apply the MSVM, and the corresponding classification result is in the last row of Table 6. Again, perfect classification is achieved in crossvalidating and testing. Indeed, the zero test error is no surprise from the picture, and we have checked that QDA (quadratic discriminant analysis), a simple traditional method, which could not be applied when the dimension of input space exceeds the sample size, gives the same zero test error once the data are represented by three PCs. Other benefit of the dimension reduction for MSVMs is that the number of SVs is noticeably reduced to about a third of the training sample size. Figure 5 shows the predicted decision vectors (f_1, f_2, f_3, f_4) at the test samples. The four class labels are coded according as EWS: $(1, -1/3, -1/3, -1/3)$, BL: $(-1/3, 1, -1/3, -1/3)$, NB: $(-1/3, -1/3, 1, -1/3)$, and RMS: $(-1/3, -1/3, -1/3, 1)$. We follow the color scheme in Figure 4, to indicate the true class identities of the test samples. For example, blue bars correspond to EWS samples, and the ideal decision vector (f_1, f_2, f_3, f_4) for them is

Table 5: 20 genes with the largest ratios in SRBCT data set

Id	Gene	Description
770394	*Fc fragment of IgG, receptor, transporter, alpha*	Regulate serum IgG level.
796258	*Sarcoglycan, alpha (50 kDa dystrophin-associated glycoprotein)*	Structure.
784224	*Fibroblast growth factor receptor 4*	Bind both acidic and basic FGF.
814260	*Follicular lymphoma variant translocation 1 (FVT-1)*	
295985	Unknown	Unknown.
377461	*Caveolin 1, caveolae protein, 22 kDa*	Structural component of caveolae.
859359	*Quinone oxidoreductase homolog*	Metabolism.
769716	*Neurofibromin 2 (bilateral acoustic neuroma)*	Possible tumor suppressor[1].
365826	*Growth arrest-specific 1*	Growth regulation.
1435862	*MIC2 (CD99)*	Transmembrane glycoprotein and tumor marker[2].
866702	*Protein tyrosine phosphatase, non-receptor type 13 (APO-1/CD95 (Fas)-associated phosphatase)*	May involve in Fas-mediated apoptosis.
296448	*Insulin-like growth factor 2 (somatomedin A)*	Growth regulation.
740604	*Interferon stimulated gene (20 kDa) (ISG20)*	
241412	*E74-like factor 1 (ets domain transcription factor) (Elf-1)*	Ets family transcription factor.
810057	*Cold shock domain protein A*	Probable transcriptional factor for Y-box.
244618	Unknown	Unknown.
52076	*Olfactomedin related ER localized protein*	
21652	*Catenin (cadherin-associated protein), alpha 1 (102 kDa)*	Structure.
43733	*Glycogenin 2 (GYG2)*	lycogen synthesis.
236282	*Wiskott-Aldrich syndrome protein (WASP)*	Related to X-linked immunodeficiency.

[1] Zhu and Parada (2001)

[2] Weidner and Tjoe (1994), Ramani et al. (1993), and Fellinger et al. (1992).

Table 6: LOOCV error and Test error for SRBCT data set. MSVMs with the Gaussian kernel are applied to the training data set. The second column indicates the optimal tuning parameters pair, λ and σ on log 2 scale chosen by the GACV. The third column presents the number of Support Vectors in the final solution of the MSVM with the number of genes specified as in the first column. The last row shows the results by using only three principal components (PCs) from 100 genes.

Number of genes	$(\log_2 \lambda, \log_2 \sigma)$	Number of SVs	LOOCV error	Test error
20	(-22,1.4)	45	0	0
60	(-23,2.4)	63	0	0
100	(-23,2.6)	58	0	0
all	(-25,4.8)	63	1	0 to 3
3 PCs (100)	(-19,1.6)	22	0	0

$(1, -1/3, -1/3, -1/3)$. The estimated decision vectors are pretty close to the ideal representation and their maximum components are the first one, meaning correct classification. We can see from the plot that all the 20 test examples from 4 classes are classified correctly. Note that the test examples are rearranged in the order of EWS, BL, NB, RMS, and non SRBCT, so the horizontal coordinates do not match with the test id's given in the original data set. In the test data set, there are 5 non SRBCT samples (2 normal muscle tissues and 3 cell lines). The fitted MSVM decision vectors for the 5 samples are plotted in cyan color in Figure 5. In clinical settings, it is important to be able to reject classification whenever samples not falling into the known classes are given. Now, we demonstrate that the MSVM predictions are specific enough to identify the peculiarity of the 5 non SRBCTs. The loss function at $\mathbf{x}$ in (2) is used to measure the MSVM prediction strength at unseen example $\mathbf{x}$, as described earlier. The smaller loss, the stronger prediction. The last panel in Figure 5 depicts the loss for the predicted MSVM decision vector at each test sample including 5 non SRBCTs. The dotted line indicates the threshold of rejecting a prediction given the loss. That is, any prediction with loss above the dotted line will be rejected. It is set at 0.2171, which is a jackknife estimate of the 95th percentile of the loss distribution from the training data set. Note that the LOOCV error for the training data set was zero, so all the 63 losses that the jackknife estimate is based on are from correct predictions. The losses corresponding to the predictions of 5 non SRBCTs all exceed the threshold, while 3 test samples out of 20 can not be classified confidently by thresholding.

4 Discussion

In this paper, we applied MSVMs to diagnose multiple cancer types based on gene expression profiles in conjunction with a method of thinning out genes. The main motivation was to demonstrate that the MSVM specially developed for multiclass problems can accurately classify cancer types and to compare the performance of MSVMs to those reported in the previous studies, in particular, that of a series of binary SVMs for multiple cancer types. The MSVM method could achieve perfect classification or near perfect classification for the two data sets considered. The classification accuracy is comparable to other methods.

The cancer classification problems using gene expression profiles available so far are observed to be very separable and not a challenging task once the dimension of the input space is reduced. In other words, it has the implication that gene expression profiles are informative enough to differen-

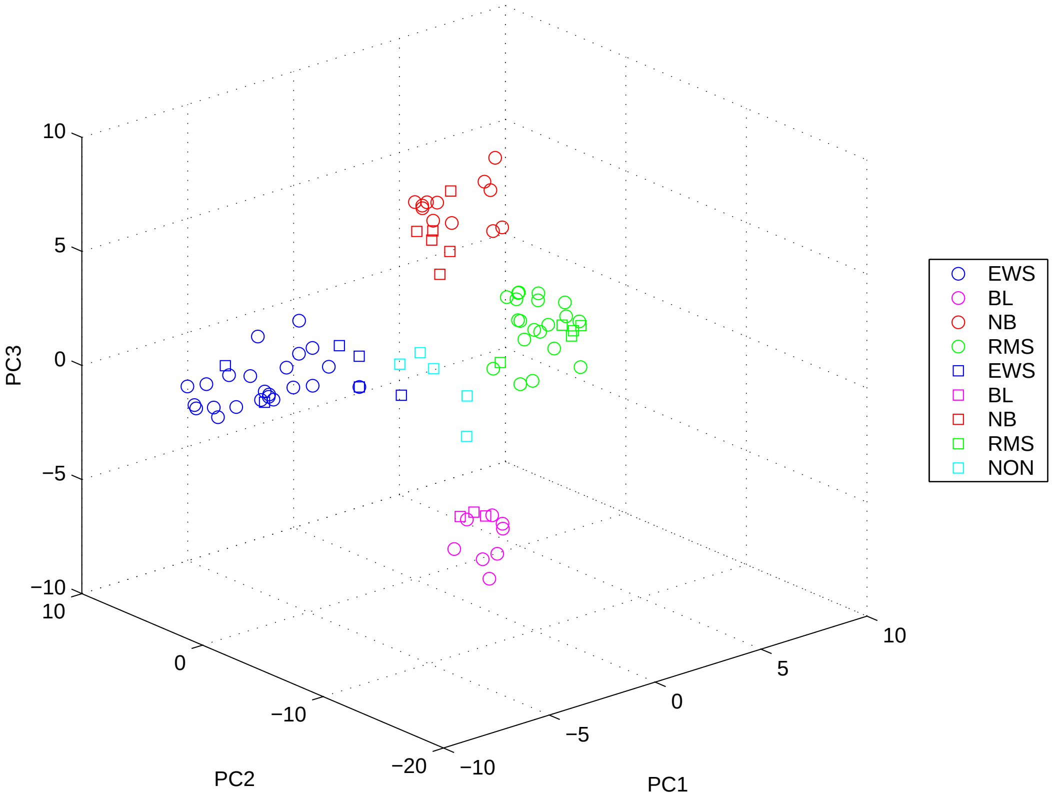

Figure 4: Three principal components of 100 gene expression levels in the training set are plotted as circles. The squares represent the corresponding principal coordinates of the test samples including non SRBCT samples. The tumor types are distinguished by colors (EWS: blue, BL: purple, NB: red, RMS: green, and non SRBCT: cyan). We can see a nice separation of the four tumor types through three principal components. Non SRBCT samples lie amid four-class samples.

tiate several tumor types. If this is a prevalent characteristic of the cancer diagnosis problem with gene expressions, then the accuracy of any reasonable classifier may not be significantly different. Differences, if any, will get evident as we accumulate more information on this kind of data. Still, there are certain advantages of flexible classifiers, for example, the SVM is often advocated to be suitable for the tumor classification problems using gene expression data, since in principle it can handle variables even larger than the sample size via its dual formulation. In this sense, MSVMs and binary SVMs may look resistant to the curse of dimensionality, but it is obvious that the presence of irrelevant noise variables does deteriorate their classification accuracy. Not only for the sake of the parsimony, the dimension reduction including gene selection is indispensable to improve accuracy.

The MSVM methodology is a generic approach to multiclass problems treating all the classes simultaneously. Solving a series of binary problems instead, in one-vs-rest fashion has potential drawbacks when classes overlap considerably. The pairwise approach often exhibits large variability since each binary classifier is estimated from a small subset of the training data. If there are quite a few classes and pooling some classes into a hyperclass gives much simpler classification problems than the original problem, the simultaneous approach may not be very efficient. Nonetheless, the difficulty is how to form such hyperclasses cleverly without a priori knowledge about the relation

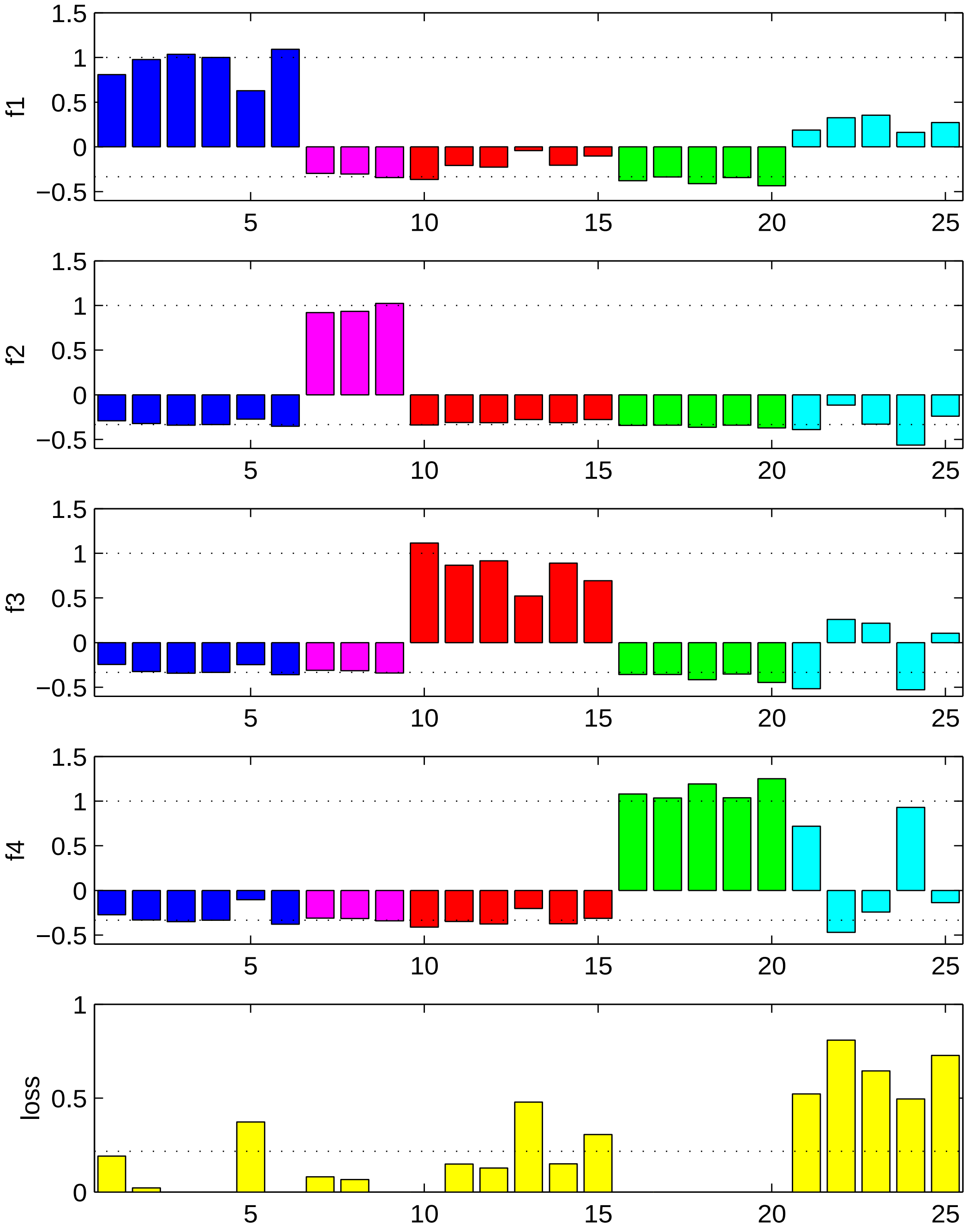

Figure 5: The first four panels show the predicted decision vectors (f_1, f_2, f_3, f_4) at the test samples. The four class labels are coded according as EWS in blue: $(1, -1/3, -1/3, -1/3)$, BL in purple: $(-1/3, 1, -1/3, -1/3)$, NB in red: $(-1/3, -1/3, 1, -1/3)$, and RMS in green: $(-1/3, -1/3, -1/3, 1)$. The colors indicate the true class identities of the test samples. We can see from the plot that all the 20 test examples from 4 classes are classified correctly and the estimated decision vectors are pretty close to their ideal class representation. The fitted MSVM decision vectors for the 5 non SRBCT samples are plotted in cyan. The last panel depicts the loss for the predicted decision vector at each test sample. The last 5 losses corresponding to the predictions of non SRBCTs all exceed the threshold (the dotted line) below which means a strong prediction. Three test samples falling into the known four classes can not be classified confidently by the same threshold.

among classes. The effectiveness of the various approaches to solve multiple tumor types diagnosis problems remains to be addressed as we collect more evidence. From the classification accuracy shown in this paper and its flexibility, we believe that the MSVM has a great potential to be used for medical diagnosis.

Note added

A few references on other extensions of the binary SVM to the multiclass case have been added and a couple of notations have been clarified on May 21, 2002. Section 2.2 has been slightly revised on July 7, 2002.

Acknowledgements

The first author would like to thank Grace Wahba and Yi Lin for their helpful suggestions and discussions.

References

Bain, G. and Murre, C. (1998). The role of E-proteins in B- and T-lymphocyte development, *Semin Immunol* **10**: 143–153.

Boser, B., Guyon, I. and Vapnik, V. (1992). A training algorithm for optimal margin classifiers, *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, Vol. 5, pp. 144–152.

Burges, C. (1998). A tutorial on support vector machines for pattern recognition, *Data Mining and Knowledge Discovery* **2**(2): 121–167.

Charrasse, S., Mazel, M., Taviaux, S., Berta, P., Chow, T. and Larroque, C. (1995). Characterization of the cDNA and pattern of expression of a new gene over-expressed in human hepatomas and colonic tumors, *Eur J Biochem* **234**: 406–413.

Crammer, K. and Singer, Y. (2000). On the learnability and design of output codes for multiclass problems, *Computational Learing Theory*, pp. 35–46.

Dudoit, S., Fridlyand, J. and Speed, T. (2000). Comparison of discrimination methods for the classification of tumors using gene expression data, *Technical Report 576*, Department of Statistics, University of California, Berkeley. J. Am. Stat. Assoc., 97(457):77–87, 2002.

Evgeniou, T., Pontil, M. and Poggio, T. (1999). A unified framework for regularization networks and support vector machines, *Technical Report AI Memo 1654*, MIT.

Fellinger, E. J., Garin-Chesa, P., Glasser, D. B., Huvos, A. G. and Rettig, W. J. (1992). Comparison of cell surface antigen HBA71 (p30/32MIC2), neuron-specific enolase, and vimentin in the immunohistochemical analysis of Ewing's sarcoma of bone., *Am J Surg Pathol* **16**: 746–755.

Furey, T., Cristianini, N., Duffy, N., Bednarski, D., Schummer, M. and Haussler, D. (2000). Support vector machine classification and validation of cancer tissue samples using microarray expression data, *Bioinformatics* **16**(10): 906–914.

Golub, T., Slonim, D., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J., Coller, H., Loh, M., Downing, J., Caligiuri, M., Bloomfield, C. and Lander, E. (1999). Molecular classification of cancer: class discovery and class prediction by gene expression monitoring, *Science* **286**: 531–537.

Hromas, R., Collins, S., Raskind, W., Deaven, L. and Kaushansky, K. (1991). Hem-1, a potential membrane protein, with expression restricted to blood cells, *Biochim Biophys Acta* **1090**: 241–244.

Khan, J., Wei, J., Ringner, M., Saal, L., Ladanyi, M., Westermann, F., Berthold, F., Schwab, M., Atonescu, C., Peterson, C. and Meltzer, P. (2001). Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks, *Nature Medicine* **7**: 673–679.

Lee, Y., Lin, Y. and Wahba, G. (2001). Multicategory Support Vector Machines, *Proceedings of the 33rd Symposium on the Interface.* Also available as University of Wisconsin-Madison Statistcs Dept TR 1043, `http://www.stat.wisc.edu/~wahba -> TRLIST`.

Lee, Y., Lin, Y. and Wahba, G. (2002). Multicategory support vector machines, theory, and application to the classification of microarray data and satellite radiance data, in preparation.

Lin, Y., Lee, Y. and Wahba, G. (2002). Support vector machines for classification in nonstandard situations, *Machine Learning* **46**: 191–202.

Mukherjee, S., Tamayo, P., Slonim, D., Verri, A., Golub, T., Mesirov, J. and Poggio, T. (1999). Support vector machine classification of microarray data, *Technical Report AI Memo 1677*, MIT.

Ramani, P., Rampling, D. and Link, M. (1993). Immunocytochemical study of 12E7 in small round-cell tumours of childhood: an assessment of its sensitivity and specificity., *Histopathology* **23**: 557–561.

Vapnik, V. (1995). *The Nature of Statistical Learning Theory*, Springer Verlag, New York.

Vapnik, V. (1998). *Statistical Learning Theory*, Wiley, New York.

Wahba, G. (1990). *Spline Models for Observational Data*, Series in Applied Mathematics, Vol. 59, SIAM, Philadelphia.

Wahba, G. (1998). Support vector machines, reproducing kernel Hilbert spaces, and randomized GACV, *in* B. Schoelkopf, C. J. C. Burges and A. J. Smola (eds), *Advances in Kernel Methods: Support Vector Learning*, MIT Press, pp. 69–87.

Wahba, G., Lin, Y. and Zhang, H. (2000). GACV for support vector machines, or, another way to look at margin-like quantities, *in* A. J. Smola, P. Bartlett, B. Scholkopf and D. Schurmans (eds), *Advances in Large Margin Classifiers*, MIT Press, pp. 297–309.

Weidner, N. and Tjoe, J. (1994). Immunohistochemical profile of monoclonal antibody O13: antibody that recognizes glycoprotein p30/32MIC2 and is useful in diagnosing Ewing's sarcoma and peripheral neuroepithelioma., *Am J Surg Pathol* **18**: 486–494.

Weston, J. and Watkins, C. (1999). Support vector machines for multiclass pattern recognition, *Proceedings of the Seventh European Symposium On Artificial Neural Networks.*

Yeo, G. and Poggio, T. (2001). Multiclass classification of SRBCTs, *Technical Report AI Memo 2001-018 CBCL Memo 206*, MIT.

Zhu, Y. and Parada, L. F. (2001). Neurofibromin, a tumor suppressor in the nervous system., *Exp Cell Res* **264**: 19–28.